

Docker 101

frank

#whoami

frank

National Chung Cheng University

- Electrical Engineering Department
- System Administrator @ DormNet

National Chiao Tung University

- Institute of Computer Science and Engineering
- Speed Lab
- Tool man @ CS Computer Center

What is docker?

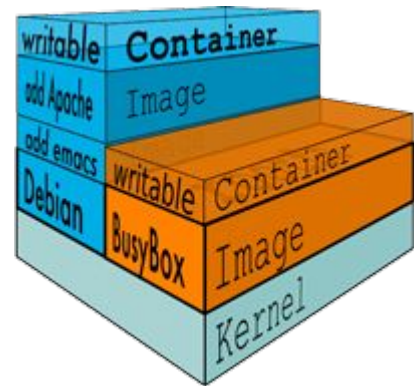
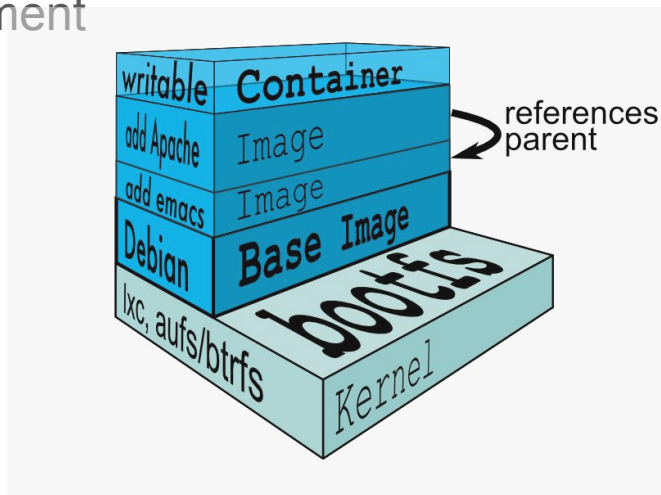
A Container:

Just like FreeBSD Jails, Solaris Zones, Linux LXC

But more...

Docker Hub

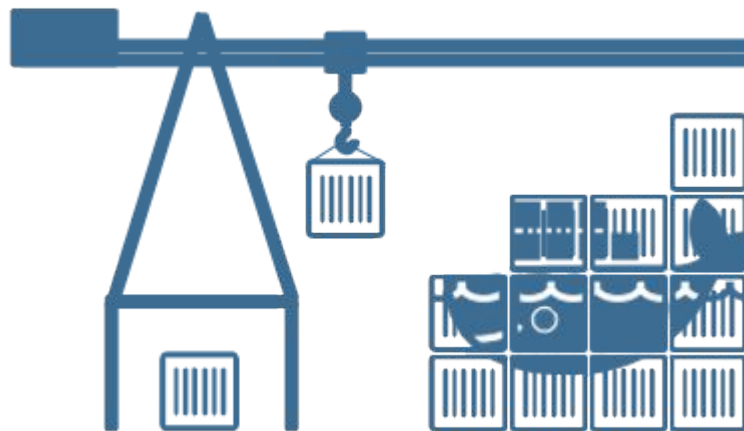
Magical image management



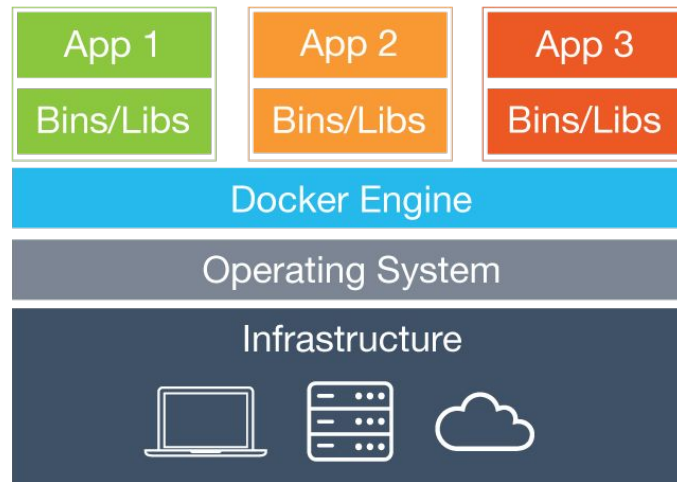
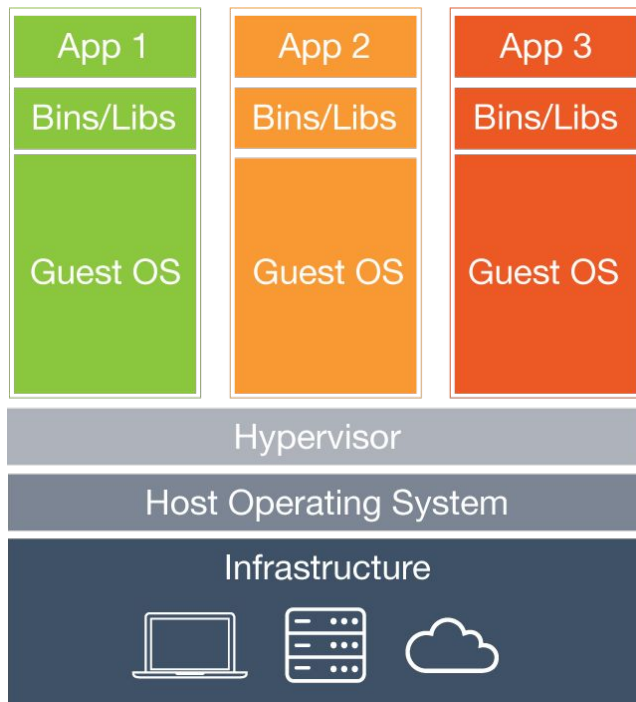
Docke Hub

A cloud docker registry with

- Public and private Docker
- Official repository
- Automated build
- Webhooks
- Collaboraters, Organization and groups



Docker v.s. VM



Why docker?

Configure once, run anywhere

Lightweight

Isolation

SCALE REAL QUICK

Docker Installation

In Ubuntu:

```
apt-get install -y docker.io
```

In Archlinux:

```
pacman -Sy docker
```

Start Docker

In Ubuntu:

```
service docker.io start
```

In Archlinux:

```
systemctl start docker
```


Deploy Docker REAL Quick

docker-machine:

create docker on your own computer

and cloud provider in a super easy manner

```
# docker-machine create --driver virtualbox docker1
```

Docker Basics - CLI

- docker
 - both client and daemon

- docker daemon
 - -H [HOST]: e.g, -H tcp://0.0.0.0:2375 -H unix://path/to/socket
 - --dns: set /etc/resolv.conf
 - --bip: cidr to use
 - --icc: container inter-connection
 - ...

Docker Basics - CLI

- Docker client
 - docker version
 - docker info
 - docker search [keyword]
 - docker push/pull/commit
 - docker run
 - --name: container name
 - -p [host port:container port] : publish a container's port
 - -P: publish all container's ports
 - docker ps
 - -a: list all container including inactive ones
 - docker build
 - -t: add tag
 - docker start/stop/attach

Basic Docker Workflow

```
# docker pull ubuntu:14.04
```

```
# docker run -it -v $HOME/.Xauthority:/root/.Xauthority \  
ubuntu:14.04
```

... make some changes ...

```
# docker commit <container id> wine
```

```
# docker push
```

Demo

with picachu

Start picachu

```
docker run --rm \  
-v $HOME/.Xauthority:/root/.Xauthority \  
--net=host \  
-e DISPLAY=$DISPLAY \  
--device /dev/snd:/dev/snd frank/picachu:v3
```

Dockerfile

Just like Makefile

Format:

Comment

INSTRUCTION arguments

Dockerfile

FROM <image>:<tag>

Set the base image, must be the first line of dockerfile

MAINTAINER <name>

Declare the maintainer of the image

RUN <command> (run in shell, i.e. sh -c <command>)

RUN ['executable', 'arg1', 'arg2'] (exec)

Dockerfile

ADD <src> <dest>

Add a file from <src> to <dest>

CMD command arg1 arg2 (use shell)

CMD ['executable', 'arg1', 'arg2',...] (exec)

CMD ['arg1', 'arg2',...] (use with ENTRYPOINT)

Command to run upon start (can be override during docker run)

ENTRYPOINT ['executable', 'arg1', 'arg2',...]

ENTRYPOINT command arg1 arg2

Dockerfile for PICACHU

```
FROM ubuntu:14.04
```

```
RUN sed -i 's/archive.ubuntu/tw.archive.ubuntu/g' /etc/apt/sources.list && \  
    dpkg --add-architecture i386 && \  
    apt-get update && \  
    apt-get install -y wine winetricks
```

```
ADD picachu.exe /root/picachu.exe
```

```
MAINTAINER frank <frank@urcrazy.net>
```

```
ENTRYPOINT wine /root/picachu.exe
```

Using Multi-container

Use `--link` to link individual container

```
docker run -d --name mydb \  
-e MYSQL_ROOT_PASSWORD=test \  
mariadb:latest
```

```
docker run -d --name wordpress \  
-p 80:80 --link mydb:mysql \  
wordpress:latest
```

Using Multi-container with docker-compose

docker-composer.yml:

web:

image: wordpress:latest

ports:

- "80:80"

links:

- db:mysql

db:

image: mariadb:latest

environment:

MYSQL_ROOT_PASSWORD: test

Docker Swarm

Create key value store using Consul:

```
docker-machine create \  
  -d virtualbox \  
  kv-store
```

```
docker $(docker-machine config kv-store) run -d \  
  -p 8500:8500 -h consul \  
  progrium/consul  
  -server -bootstrap
```

Docker Swarm

Create swarm master:

```
docker-machine create \  
  -d virtualbox \  
  --swarm \  
  --swarm-master \  
  --swarm-discovery="consul://$(docker-machine ip kv-store):8500" \  
  --engine-opt="cluster-store=consul://$(docker-machine ip kv-store):8500" \  
  --engine-opt="cluster-advertise=eth1:2376" \  
  node1-master
```

Docker Swarm

Create swarm node:

```
docker-machine create \  
  -d virtualbox \  
  --swarm \  
  --swarm-discovery="consul://$(docker-machine ip kv-store):8500" \  
  --engine-opt="cluster-store=consul://$(docker-machine ip kv-store):8500" \  
  --engine-opt="cluster-advertise=eth1:2376" \  
node2
```

Docker Multi-hosting Networking (starting from 1.9)

```
eval $(docker-machine env --swarm node1-master)
```

```
docker network create --driver overlay mynet
```

```
docker run --name c1 -it -e constraint:node==node1-master \  
--net=mynet busybox
```

```
docker run --name c2 -it -e constraint:node==node2 \  
--net=mynet busybox
```


Happy dockering
Thanks